Package ‘FBCanalysis’

March 28, 2022

Title Develop and evaluate time series data models based on fluctuation based clustering

Version 0.0.0.9000

Date 2020-03-16

Description The package includes tools for performing fluctuation-based clustering (FBC) on biological time series data, primarily for monitoring data fluctuations in asthmatics. The package includes functions for registering and processing time series data, developing matching clustering models based on Earth Mover's distances, and evaluating the models through enrichment analysis, stability after random data removal, or other frequently used cluster stability metrics.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

URL `https://github.com/MrMaximumMax/FBCanalysis`

BugReports `https://github.com/MrMaximumMax/FBCanalysis/issues`

Depends R (>= 4.1.1)

Imports arsenal,
cluster,
clValid,
dplyr,
emdist,
FCPS,
glmnet,
imputeTS,
lubridate,
mclust,
RankAggreg,
readr,
stats,
tibble,
utils

Suggests rmarkdown,
knitr

VignetteBuilder knitr

R topics documented:

`add_clust2enrich` *Add clustering assignments from clustering output to preprocessed enrichment data frame.*

Description

Add clustering assignments from clustering output to preprocessed enrichment data frame.

Usage

```
add_clust2enrich(enrich, clustdat)
```

Arguments

`enrich`	Preprocessed enrichment data frame (also see function: add_enrich)
`clustdat`	Object of type list storing clustering data (also see function: clust_matrix)

Value

Processed enrichment data frame with added column indicating cluster assignments

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
clustering <- clust_matrix(matrix, method = "kmeans", nclust = 3)
enr <- add_enrich(list,
'https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/enrich/enrichment.csv')
enr <- add_clust2enrich(enr, clustering)
```

`add_clust2ts` *Add clustering assignments from clustering output to time series data list and store the data in a data frame.*

Description

Add clustering assignments from clustering output to time series data list and store the data in a data frame.

Usage

```
add_clust2ts(plist, clustdat)
```

Arguments

`plist`	List storing patient time series data (also see function: patient_list)
`clustdat`	Object of type list storing clustering data (also see function: clust_matrix)

Value

Processed data frame storing time series data with added column indicating cluster assignments

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
clustering <- clust_matrix(matrix, method = "kmeans", nclust = 3)
ts <- add_clust2ts(list, clustering)
```

add_enrich *Add enrichment data from a csv file, match with Patient ID entries from a previously generated time series data list and preprocess for further analysis.*

Description

Add enrichment data from a csv file, match with Patient ID entries from a previously generated time series data list and preprocess for further analysis.

Usage

```
add_enrich(plist, path)
```

Arguments

plist	List storing patient time series data (also see function: patient_list)
path	Path where enrichment csv file is stored

Details

the enrichment csv file should have a column including the Patient ID. Additionally, the user specifies the list in which time series data is saved. This is advantageous since the function can now do matching, i.e. determine which Patient IDs occur in both the enrichment dataset and time series datalist. So for a result, any Patient ID that appears in the enrichment dataset but does not exist in the time series datalist will be deleted from the enrichment dataset, as it cannot be used in any further investigation. Nonetheless, Patient IDs from the time series data that do not appear in the enrichment dataset will be added to the enrichment dataset, but each new parameter will be featureless, so added as NA value.

If the user selects option 1 (leave missing values as NA), no further processing of the input occurs. The enrichment data set will be added to the environment as a data frame. In this situation, the NA values from the enrichment dataset will be included in the summary indicating, for example, that a certain cluster has a given percentage of missing values. This may also lead to some additional findings, such as that a specific parameter considerably enriches a cluster yet many data is absent. If the user selects option 2 (sample missing values), the function loops over each NA entry and selects a random value from the whole distribution for the parameter for which the data is missing. This cycle is repeated until the whole dataset has been processed and the data will be added as a data frame to the environment.

Value

Processed data as object of type data frame; Enrichment data Patient_IDs are matched with Time Series Data List Patient IDs; In case it was indicated, NA values in the enrichment data are filled up by random sampling

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
```

```
enr <- add_enrich(list,
'https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/enrich/enrichment.csv')
```

clust_matrix

Cluster Earth Mover Distance Square Matrix data and record cluster assignments for involved Patient_IDs for a specified clustering technique and number of clusters.

Description

Cluster Earth Mover Distance Square Matrix data and record cluster assignments for involved Patient_IDs for a specified clustering technique and number of clusters.

Usage

```
clust_matrix(matrix, method, nclust, plotclust)
```

Arguments

`matrix`	Object of type matrix storing Earth Mover's Distances for patient time series data distribution pairs
`method`	Clustering method (hierarchical, kmeans, diana, fanny, som, modelbased, sota, pam, clara)
`nclust`	Number of clusters (if not specified, user will be asked in the terminal)
`plotclust`	TRUE/FALSE if clustering data should be visualized (TRUE by default)

Details

Hierarchical clustering describes a general agglomerative hierarchical clustering approach in which the optimum value of an objective function is used to choose which pair of clusters should merge at each step (see hclust for more details).

K-means clustering is a vector quantization technique that divides a set of n observations into k groups, with each observation belonging to the cluster with the closest mean or centroid (see kmeans for more details).

The divisive analytic method (DIANA) constructs a hierarchical clustering structure, starting with a single huge cluster containing all n observations. Clusters are further split until each has a single observation. At each step, the cluster with the largest diameter is selected, where the diameter of a cluster is defined as the biggest difference between any two of its observations (see diana for more details).

While partitioning around medoids (PAM) is comparable to k-means, it is considered more robust since it allows for the use of dissimilarities other than euclidean distance. As with k-means, the number of clusters is determined in advance, and an initial set of cluster centers is required to begin the process (see pam for more details).

Clustering large applications (CLARA) is a system that involves sampling to apply PAM to a sequence of sub-datasets. When the number of observations is big, this results in shorter run times. It is substantially faster than other partitioning algorithms such as PAM at handling huge datasets. Internally, this is performed by examining fixed-size sub-datasets, which results in linear rather than quadratic time and storage requirements for n (see clara for more details).

FANNY explains a fuzzy analysis clustering method. Each observation is distributed through-out the numerous groups in a fuzzy clustering (see fanny for more details).

Self-organizing maps (SOM) are a widespread unsupervised learning technique used by com- putational biologists and academics in machine learning. SOM is a neural network-based system that is well-known for its ability to map and display two-dimensional data (see SOMclustering for more details).

Modelbased clusterinng fits the data to a statistical model composed of a finite mixture of Gaussian distributions. Each mixture component represents a cluster, and the maximum like- lihood method, or estimation maximum (EM), is used to determine the mixture components and group memberships (see ModelBasedClustering for more details).

SOTA, self-organizing tree algorithm, denotes an unsupervised network with a hierarchical and divisive binary tree topology. It is a fast approach, which makes it suitable for clustering a large number of elements. It combines the advantages of hierarchical clustering with those of SOMs. The algorithm chooses the most diverse node and separates it into two nodes referred as cells (see sota for more details).

Value

Object of type list storing cluster data and clustering assignments for the Patient_IDs from the Earth Mover's Distance matrix

References

Joe H Ward Jr. Hierarchical grouping to optimize an objective function. Journal of the American statistical association, 58(301):236–244, 1963.

Stephane Tuffery. Data mining and statistics for decision making. John Wiley & Sons, 2011.

Fionn Murtagh. A survey of recent advances in hierarchical clustering algorithms. The computer journal, 26(4):354–359, 1983.

Fionn Murtagh. Clustering in massive data sets. In Handbook of massive data sets, pages 501–543. Springer, 2002.

Greg Hamerly and Charles Elkan. Alternatives to the k-means algorithm that find better clusterings. In Proceedings of the eleventh international conference on Information and knowledge management, pages 600–607, 2002.

R Wehrens and J Kruisselbrink. kohonen: Supervised and unsupervised self-organising maps r package version 3.0. 10, 2019.

Leonard Kaufman and Peter J Rousseeuw. Finding groups in data: an introduction to cluster analysis. John Wiley & Sons, 2009.

Mark Van der Laan, Katherine Pollard, and Jennifer Bryan. A new partitioning around medoids algorithm. Journal of Statistical Computation and Simulation, 73(8):575–584, 2003.

Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. Journal of the American statistical Association, 97(458):611–631, 2002.

Javier Herrero, Alfonso Valencia, and Joaquın Dopazo. A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics, 17(2):126–136, 2001.

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
```

```
matrix <- emd_matrix(list, "FEV1")
clustering <- clust_matrix(matrix, method = "hierarchical", nclust = 2)
```

clValid_flow — *Interactive console workflow to calculate and evaluate cluster validation measures which have been determined previously by the call init_clValid.*

Description

Interactive console workflow to calculate and evaluate cluster validation measures which have been determined previously by the call init_clValid.

Usage

```
clValid_flow(matrix, par)
```

Arguments

`matrix`	Earth Mover's Distance Matrix for processed patient time series data (also see functions: emd_matrix, patient_list)
`par`	Object of type list storing clustering methods and cluster range of interest; initialized via function: init_clValid

Details

The call guides through an interactive workflow and generates cluster evaluation measures, stores and lists, visualizes corresponding plots and lets the user decide which technique is the prefered one. Once the user has chosen his favourite, the flow continues to the function clust_matrix and generates the respective clustering output. The internal cluster validation methods utilize just the dataset and the clustering partition as input and evaluates the clustering's quality by using intrinsic information included in the data.

The call calculates Connectivity, Silhouette width and Dunn index. Connectivity describes the connectness to neighbors of particular clustering partition and should be minimized. Silhouette width defines the average silhouette value for each observation and should be maximized. The Dunnn index is a definition for Ratio of shortest distance between non-cluster observation and greatest intra-cluster distance and should be maximized likewise.

Furthermore, cluster stability measures are available, namely Average proportion of non-overlap (APN), Average distance (AD), Average distance between means (ADM) and Figure of merit (FOM). APN is the average proportion of observations that are not clustered using complete and leaky data. AD defines the average distance in observations for both complete and leaky data. ADM deals with the average distance between cluster centers in complete and leaky data. FOM is a measure for average intra- cluster variance in leaky data. All measures should be minimized. Furthermore, Rank Aggregation may be performed. It approaches to provide a generic and flexible framework for objectively integrating several ordered lists in a suitable and efficient way. The used technique for evaluating clustering the rank is by a Cross-entropy approach, which is incorporating Spearman's footrule distance measure. In the end a recommendation for the best fitting clustering model is given.

Value

Object of type list storing chosen clustering method and number of clusters (can be then used for function clust_matrix)

References

Guy Brock, Vasyl Pihur, Susmita Datta, and Somnath Datta. clvalid: An r package for cluster validation. Journal of Statistical Software, 25:1–22, 2008.

Julia Handl, Joshua Knowles, and Douglas B Kell. Computational cluster validation in post-genomic data analysis. Bioinformatics, 21(15):3201–3212, 2005.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20:53–65, 1987.

Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. Journal of cybernetics, 4(1):95–104, 1974.

Vasyl Pihur, Susmita Datta, and Somnath Datta. Weighted rank aggregation of cluster validation measures: a monte carlo cross-entropy approach. Bioinformatics, 23(13):1607–1615, 2007.

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
distmat <- emd_matrix(list, "PEF", maxIter = 5000)
parameters <- init_clValid()
output <- clValid_flow(distmat, parameters)
```

emd_heatmap *Visualize an Earth Mover's Distance Square Matrix as a heatmap*

Description

Visualize an Earth Mover's Distance Square Matrix as a heatmap

Usage

```
emd_heatmap(input, parameter, maxIter)
```

Arguments

input	Earth Mover's Distance Matrix or list storing patient time series data (also see function: patient_list)
parameter	In case list is input, the parameter of interest from time series data list
Iter	In case input is time series data list, incate maxIter to calculate EMD matrix (also see function: emd_matrix)

Value

Visualized Earth Mover's Distance Matrix as a heatmap

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
matrix <- emd_matrix(list, "FEV1")
emd_heatmap(matrix)
```

---

emd_matrix *Generate an Earth Mover's Distance Matrix for time series data distributions pairs out of a preprocessed time series data list.*

Description

Generate an Earth Mover's Distance Matrix for time series data distributions pairs out of a preprocessed time series data list.

Usage

```
emd_matrix(plist, parameter, maxIter)
```

Arguments

plist	List storing patient time series data (also see function: patient_list)
parameter	Parameter of interest to determine Earth Mover's Distances between distributions
maxIter	Maximum of iterations to calculate Earth Mover's Distance (default: 500)

Details

the function may compute the EMD for each Patient ID pairi,j using the normalized distributions. EMD is a distance measure between two probability distributions over a region D. Informally, if the distributions are viewed as two distinct methods of accumulating a certain quantity of earth gravel across the region D. EMD is the smallest cost associated with converting one pile to another, where the cost is supposed to equal the quantity of material transferred multiplied by the distance traveled. A unit of labor is defined in this context as conveying a unit of earth across a unit of ground distance. A distribution may be described as a collection of clusters, each of which is defined by its mean or mode and the proportion of the distribution that belongs to it. This representation is referred to as the distribution's signature. Both signatures may be of varying sizes. Simple distributions, for example, have lower signatures than complex distributions. (also see emd for further details)

Value

Earth Mover's Distance Square Matrix of type matrix

References

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In Sixth International Conference on Computer Vi- sion (IEEE Cat. No. 98CH36271), pages 59–66. IEEE, 1998.

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
matrix <- emd_matrix(list, "FEV1")
```

enr_obs_clust *Observe a specific cluster of interest on preporcessed enrichment and time series data for overview and p-values. The p-values are calculated by a Wilcox Rank Sum test for continuous data and a Fisher's Exact Test for categorical enrichment data.*

Description

Observe a specific cluster of interest on preporcessed enrichment and time series data for overview and p-values. The p-values are calculated by a Wilcox Rank Sum test for continuous data and a Fisher's Exact Test for categorical enrichment data.

Usage

```
enr_obs_clust(ts.dat, enrich, clustno)
```

Arguments

ts.dat	Processed data frame storing time series data and cluster assignments (also see function: add_clust2ts)
enrich	Processed data frame storing enrichment data and cluster assignments (also see function: add_clust2enrich)
clustno	Cluster number of interest

Details

There are two techniques to compute the p-value for continuous or categorical data, according to past work. In order to determine the relevant p-value inside a cluster of interest, the data distribution within the cluster should be compared to the data distribution outside the cluster. Prior to conducting the related probability tests, one data processing step is performed, namely the construction of two data distributions, one including only data included inside the cluster and another comprising data from outside the cluster, for the purpose of comparing them.

The Mann-Whitney Test, sometimes referred to as the Wilcoxon rank-sum test (WRS), is used to measure the significance of continuous variables within the observed distribution. The WRS is used to test if the central tendency of two independent samples is different. When the t-test for independent samples does not meet the requirements, the WRS is used. The null hypothesis H0 states that the populations' distributions are equal. H1 is the alternative hypothesis meaning that the distributions are not equal. The test is consistent under the broader formulation only when the following happens under H1.

The hypergeometric test or Fisher's exact test (FET) is used to analyse categorical variables within the enriched data set. It is a statistical significance test for contingency tables that is employed in the study of them. The test is helpful for categorical data derived from object classification. It is used to assess the importance of associations and inconsistencies between classes. The FET is often used in conjunction with a 2 × 2 contingency table that represents two categories for a variable, as well as assignment inside or outside of the cluster. The p-value is calculated as if the table's margins are fixed. This results in a hypergeometric distribution of the numbers in the table cells under the null hypothesis of independence. A hypergeometric distribution is a discrete probability distribution that describes the probability of k successes, defined as random draws for which the object drawn has a specified feature in n draws without replacement from a finite population of size N containing exactly K objects with that feature, where each draw is either successful or unsuccessful. The test is only practicable for normal computations in the presence of a 2 × 2 contingency table. However, the test's idea may be extended to the situation of a m × n table in general. Statistics programs provide a Monte Carlo approach for approximating the more general case.

Value

Terminal output presenting summary of time series and enrichment data with corresponding p-values

References

Siegel Sidney. Nonparametric statistics for the behavioral sciences. The Journal of Nervous and Mental Disease, 125(3):497, 1957.

Kinley Larntz. Small-sample comparisons of exact levels for chi-squared goodness-of-fit statistics. Journal of the American Statistical Association, 73(362):253–263, 1978.

Cyrus R Mehta and Nitin R Patel. A network algorithm for performing fisher's exact test in r× c contingency tables. Journal of the American Statistical Association, 78(382):427–434, 1983.

Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences, 102(43):15545–15550, 2005.

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
clustering <- clust_matrix(matrix, method = "kmeans", nclust = 3)
enr <- add_enrich(list,
'https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/enrich/enrichment.csv')
enr <- add_clust2enrich(enr, clustering)
ts <- add_clust2ts(list, clustering)
enr_obs_clust(ts, enr, 1)
```

`init_clValid` *Initialize Cluster Validation Measure Analysis in the context of Fluctuation Based Clustering (FBC) analysis. The call initialized a interactive console workflow where the user may indicate the clustering techniques of interest as well as the cluster numbers of interest.*

Description

Initialize Cluster Validation Measure Analysis in the context of Fluctuation Based Clustering (FBC) analysis. The call initialized a interactive console workflow where the user may indicate the clustering techniques of interest as well as the cluster numbers of interest.

Usage

```
init_clValid()
```

Details

See clust_matrix for more details on possible clustering techniques

Value

Object of type list storing cluster method(s) and number of cluster range of interest (to be used for function: clValid_flow)

References

Guy Brock, Vasyl Pihur, Susmita Datta, and Somnath Datta. clvalid: An r package for cluster validation. Journal of Statistical Software, 25:1–22, 2008.

Examples

```
init_clValid()
```

---

jaccard_run_cognate

Simulate amount of random data removal from time series data list and determine Jaccard index via Cognate Cluster approach for multiple random data removal steps for a specific cluster of interest.

Description

Simulate amount of random data removal from time series data list and determine Jaccard index via Cognate Cluster approach for multiple random data removal steps for a specific cluster of interest.

Usage

```
jaccard_run_cognate(
  plist,
  parameter,
  n_simu,
  method,
  clust_num,
  n_clust,
  range
)
```

Arguments

`plist`	Object of type list storing patient time series data (also see function: patient_list)
`parameter`	Parameter of interest in time series data list
`n_simu`	Number of simulations
`method`	Clustering method (also see function: clust_matrix)
`clust_num`	Cluster of interest
`n_clust`	Number of clusters
`range`	Range to simulate random data removal (e.g. c(0.1,0.2,0.5,0.7,0.8))

Details

See sim_jaccard_cognate for more detailed approach on Jaccard index determination. The difference in this function is that now only one cluster is observed für multiple amoiunts of random data removal where for each data removal step defined the resulting Jaccard indices are stored in a list object. Furthermore, a boxplot visualization is generated, in the style of recent publications.

Value

Object of type list storing Jaccard indices for each indicated random data removal step and visualized results in a boxplot

References

Anja Jochmann, Luca Artusio, Hoda Sharifian, Angela Jamalzadeh, Louise J Fleming, Andrew Bush, Urs Frey, and Edgar Delgado-Eckert. Fluctuation-based clustering reveals phenotypes of patients with different asthma severity. ERJ open research, 6(2), 2020.

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
output <- jaccard_run_cognate(list,"PEF",10,"hierarchical",1,3,c(0.005,0.01,0.05,0.1,0.2))
```

`jaccard_run_emd`

Simulate amount of random data removal from time series data list and determine Jaccard index via Earth Mover's Distance approach for multiple random data removal steps for a specific cluster of interest.

Description

Simulate amount of random data removal from time series data list and determine Jaccard index via Earth Mover's Distance approach for multiple random data removal steps for a specific cluster of interest.

Usage

```
jaccard_run_emd(plist, parameter, n_simu, method, clust_num, n_clust, range)
```

Arguments

`plist`	Object of type list storing patient time series data (also see function: patient_list)
`parameter`	Parameter of interest in time series data list
`n_simu`	Number of simulations
`method`	Clustering method (also see function: clust_matrix)
`clust_num`	Cluster of interest
`n_clust`	Number of clusters
`range`	Range to simulate random data removal (e.g. c(0.1,0.2,0.5,0.7,0.8))

Details

See sim_jaccard_emd for more detailed approach on Jaccard index determination. The difference in this function is that now only one cluster is observed für multiple amoiunts of random data removal where for each data removal step defined the resulting Jaccard indices are stored in a list object. Furthermore, a boxplot visualization is generated, in the style of recent publications.

Value

Object of type list storing Jaccard indices for each indicated random data removal step and visualized results in a boxplot

References

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271), pages 59–66. IEEE, 1998.

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
output <- jaccard_run_emd(list,"PEF",10,"hierarchical",1,3,c(0.005,0.01,0.05,0.1,0.2))
```

`max_fluc` *Determine the pair of maximum fluctuation difference on time series data distribution from a preprocessed list or Earth Mover's Distance square matrix.*

Description

Determine the pair of maximum fluctuation difference on time series data distribution from a preprocessed list or Earth Mover's Distance square matrix.

Usage

```
max_fluc(input, parameter, maxIter)
```

Arguments

`input` Either a list storing time series data or EMD martrix (also see functions: patient_list, emd_matrix)

`parameter` Parameter of interest from time series data list

`maxIter` Maximum of iterations to apply for calculation of Earth Mover's Distannce (also see function: emd_matrix)

Value

Console output with Patient_ID pair, corresponding Earth Mover's Distance and visualized boxplot of both time series data distributions

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
max_fluc(list, "PEF")
```

`patient_boxplot` *Visualize patient(s) time series from a preprocessed list with in a boxplot, either as normalized or non-normalized for an indicated parameter.*

Description

Visualize patient(s) time series from a preprocessed list with in a boxplot, either as normalized or non-normalized for an indicated parameter.

Usage

```
patient_boxplot(plist, patients, parameter, normalized)
```

Arguments

`plist` List storing patient time series data (also see function: patient_list)

`patients` Patient_ID(s) referring to (a) list element; can be single ID or multiple IDs (also see function: patient_list)

`parameter` Parameter of interest in list element(s)

`normalized` TRUE/FALSE if z-normalized (TRUE by default)

Value

Visualized patient(s) time series data in a boxplot for indicated parameter

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
patient_boxplot(list,c("ID_2","testpat_1","testpat_2","a301"), "FEV1")
```

---

patient_hist *Visualize patient time series data from a preprocessed in a histogram for indicated parameter either normalized or non-normalized.*

Description

Visualize patient time series data from a preprocessed in a histogram for indicated parameter either normalized or non-normalized.

Usage

```
patient_hist(plist, Patient_ID, parameter, normalized)
```

Arguments

`plist`	List storing patient time series data (also see function: patient_list)
`Patient_ID`	Patient_ID referring to a list element (also see function: patient_list)
`parameter`	Parameter of interest in list element
`normalized`	TRUE/FALSE if z-normalized (TRUE by default)

Value

Visualized patient time series data in a histogram for indicated parameter

Examples

```
list <- patient_list("
https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
patient_hist(list,"testpat_1","PEF")
```

`patient_list` *Process patient time series data by interpolation options and store data in an object of type list.*

Description

Process patient time series data by interpolation options and store data in an object of type list.

Usage

```
patient_list(path, GitHub)
```

Arguments

`path`	Path where csv file(s) are stored (only folder, not specific file(s))
`GitHub`	Set TRUE when csv file comes form GitHub (FALSE by default); only in demo needed

Details

Prior to undertaking an analysis using one of the FBC procedures, it is necessary to adequately process and prepare the relevant time series data. The function then creates an interactive flow using the console in R Studio. To begin, the method retrieves all csv files in the provided folder, indicating that it is capable of handling multiple files. The function extracts all csv files from the given directory and merges them into a single raw data frame. The user then indicates which column represents Patient ID and time for adequate processing. The csv files are merged, columns are selected where the Patient ID column will be renamed ”Patient_ID” and the time column will be titled ”Time”. This standardization approach is critical for subsequent features because it enables the easy detection of time series data and the consistent computation and processing of data, for example z-normalization.

The user should also indicate in the interactive console the time formate which will be standardized with the help of lubridate. This is crucial because the technique can now filter the raw data by Patient ID, extract the start and end timestamps for each Patient ID, and then align the data if any records are missing while maintaining the indicated sample frequency.

The user may choose between seven approaches: L1 Regularization/Least absolute shrinkage and selection operator (LASSO) Regression, L2 Regularization/Ridge Regression, Elastic Net Regularization, Linear interpolation, Cubic C2 interpolation or, according to recent articles, fill in missing values using the highest or lowest quartile of measurements in the given time series data distribution.

The Regression and Regularization techniques generate adequate polynomials for each possible degree n-1 (where n is the total number of data points). Afterwards, cross-validation (from glmnet) is applied to determine the lambda value for the lowest MSE of the model. Afterwards, the model with polynomial degree for the lowest MSE is chosen and the missing data is interpolated with the regularized model.

It is also possible to apply a simple linear interpolation in between missing time series data points. It may be the easiest option to employ straight lines between neighboring points (also see na_interpolation). Nevertheless, these basic spline polynomials may be notoriously inexact. Cubic spline polynomials mostly provide better results.

Another option for the user is to apply the interpolation by using a cubic C spline. It implies that the composite function S must be twice continuously differentiable from all boundaries or subintervals (also see na_interpolation).

Without regression, regularization or interpolation, the user may opt to sample missing values within time series data by randomly choosing a value from the greatest or lowest quartile readings from each patient distribution. The R function then loops over each NA element in the time series data distribution of a patient for the specified parameter and randomly samples a value for the chosen quartile until the data frame is complete.

Value

Object of type list storing patient time series data

References

Jerome Friedman, Trevor Hastie, Rob Tibshirani, Balasubramanian Narasimhan, Ken- neth Tay, Noah Simon, and Junyang Qian. Package ‘glmnet’. Journal of Statistical Software. 2010a, 33(1), 2021.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. Journal of the royal statistical society: series B (statistical methodology), 67(2):301– 320, 2005.

Steffen Moritz and Thomas Bartz-Beielstein. imputets: time series missing value imputation in r. R J., 9(1):207, 2017.

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
```

patient_ts_plot *Visualize patient time series data from a preprocessed in a time series plot for an indicated parameter, either as normalized or non-normalized.*

Description

Visualize patient time series data from a preprocessed in a time series plot for an indicated parameter, either as normalized or non-normalized.

Usage

```
patient_ts_plot(plist, Patient_ID, parameter, normalized)
```

Arguments

`plist`	List storing patient time series data (also see function: patient_list)
`Patient_ID`	Patient_ID referring to a list element (also see function: patient_list)
`parameter`	Parameter of interest in list element
`normalized`	TRUE/FALSE if z-normalized (TRUE by default)

Value

Visualized patient time series data in a time series plot for indicated parameter

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
patient_ts_plot(list,"testpat_1","PEF")
```

rnd_dat_rm — *Remove a specific amount of data randomly from a time series data list.*

Description

Remove a specific amount of data randomly from a time series data list.

Usage

```
rnd_dat_rm(plist, removal)
```

Arguments

`plist`	Object of type list storing patient time series data (also see function: patient_list)
`removal`	Amount of data removal (0 = 0%, 1 = 100%)

Value

Object of type list storing patient time series data with indicated amount of data removed randomly

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
list_rm <- rnd_dat_rm(testlist, 0.95)
```

sim_jaccard_cognate — *Simulate random data removal for a removal amount with indicated number of simulations from time series data list and determine Jaccard index for all cluster via Cognate Cluster cluster assignment approach*

Description

Simulate random data removal for a removal amount with indicated number of simulations from time series data list and determine Jaccard index for all cluster via Cognate Cluster cluster assignment approach

Usage

```
sim_jaccard_cognate(plist, parameter, removal, n_simu, method, n_clust, Iter)
```

Arguments

`plist` Object of type list storing patient time series data (also see function: patient_list)

`parameter` Parameter of interest in time series data list

`removal` Amount of random data removal to determine Jaccard index

`n_simu` Number of simulations

`method` Clustering method (also see function: clust_matrix)

`n_clust` Number of clusters (also see function: clust_matrix)

`Iter` Maximum iterations to determine Earth Mover's Distances (also see function: emd_matrix); default is 5,000 for this function

Details

The cognate cluster approach works in the manner that first a Gold Standard cluster is determined meaning the cluster assignments without any data removal. Subsequently, random data is removed from the original, complete data and clustering is performed again on the leaky data. The cluster determined to be cognate to the Gold Standard cluster is the one with the highest overlap in cluster members, meaning hte cluster with highest acheived Jaccard index. Afterwards, the Jaccard indices are calculated, comparing cluster members with complete and leaky data, for each cluster.

Value

Object of type matrix storing received Jaccard indices for indicated amount of random data removal for all clusters

References

Anja Jochmann, Luca Artusio, Hoda Sharifian, Angela Jamalzadeh, Louise J Fleming, Andrew Bush, Urs Frey, and Edgar Delgado-Eckert. Fluctuation-based clustering reveals phenotypes of patients with different asthma severity. ERJ open research, 6(2), 2020.

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
output <- sim_jaccard_cognate(list, "PEF", 0.05, 10, "hierarchical", 2, 1000)
```

sim_jaccard_emd — *Simulate random data removal for a removal amount with indicated number of simulations from time series data list and determine Jaccard index for all clusters via Earth Mover's distance cluster assignment approach.*

Description

Simulate random data removal for a removal amount with indicated number of simulations from time series data list and determine Jaccard index for all clusters via Earth Mover's distance cluster assignment approach.

Usage

```
sim_jaccard_emd(plist, parameter, removal, n_simu, method, n_clust, Iter)
```

Arguments

`plist`	Object of type list storing patient time series data (also see function: patient_list)
`parameter`	Parameter of interest in time series data list
`removal`	Amount of random data removal to determine Jaccard index
`n_simu`	Number of simulations
`method`	Clustering method (also see function: clust_matrix)
`n_clust`	Number of clusters (also see function: clust_matrix)
`Iter`	Maximum iterations to determine Earth Mover's Distances (also see function: emd_matrix); default is 5,000 for this function

Details

This method represents a novel approach and potential complementary method to sim_jaccard_cognate. First, clustering is performed on complete data without removal, servinng as Gold Standard clusters. For every Gold Standard cluster then, all time series data from all patients is z-normalized and then assumed to be as one Gold Standard distribution. Subsequently, random data is removed form the time series data. Each leaky data distribution is then compared via Earth Mover's Distance to each Gold Standard Distribution. The Gold Standard cluster distribution to which the observed leaky distribution exhibits the lowest Earth Mover's Distance gets the assignment. This process is repeated until every leaky time series data distribution is assigned to a cluster. Afterwards, the Jaccard indices are calculated, comparing cluster members with complete and leaky data, for each cluster.

Value

Object of type matrix storing received Jaccard indices for indicated amount of random data removal for all clusters

References

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271), pages 59–66. IEEE, 1998.

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
output <- sim_jaccard_emd(list, "PEF", 0.05, 10, "hierarchical", 2, 100)
```

sim_sample_enr *Simulate random sampling for NA entries in enrichment data and check stability of resulting p-values for the enrichment parameters for an indicated number of random sampling simulations.*

Description

Simulate random sampling for NA entries in enrichment data and check stability of resulting p-values for the enrichment parameters for an indicated number of random sampling simulations.

Usage

```
sim_sample_enr(plist, path, clustdat, clustno, n_sim)
```

Arguments

plist	List storing patient time series data (also see function: patient_list)
path	Path where enrichment csv file is stored
clustdat	Object of type list storing clustering data (also see function: clust_matrix)
clustno	Cluster number of interest
n_sim	Number of simulations

Details

It allows the sampling in NA entries to be repeated for each parameter in the enriched data set. The primary objective here is to validate the random sampling process for missing data by running many simulations and comparing the resultant p-values. An enrichment data frame with NA elements is saved as a simulation foundation. This data frame will always serve as the foundation for any subsequent simulations added. Following that, the program runs through each NA item in the dataset and generates a random sample of the current parameter's distribution. After completing this step for each parameter, the function generates the associated p-values as explained in enr_obs_clust.

Value

Object of type list storing the received p-values for each parameter in a vector and boxplot visualizing the received p-values

Examples

```
list <- patient_list(
"https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/phys/data.csv",
GitHub = TRUE)
#Sampling frequency is supposed to be daily
path <- 'https://raw.githubusercontent.com/MrMaximumMax/FBCanalysis/master/demo/enrich/enrichment.csv'
test <- sim_sample_enr(list,path,clustering,1,100)
sim_sample_enr <- function(plist, path, clustdat, clustno, n_sim)
```

znorm *z-normalise data*

Description

Function applicable on any numeric distribution of data and z-normalize them.

Usage

```
znorm(data)
```

Arguments

data	Numeric distribution; may be stored in an object of type vector, matrix or data frame

Value

z-normalized distribution of input

Examples

```
random_distribution <- runif(n = 50, min = 1, max = 10)
znorm_distribution <- znorm(random_distribution)
```

Index